PelicanHPC Tutorial

September 2011
Michael Creel
Universitat Autònoma de Barcelona

You can check for more recent versions of this document at
http://pelicanhpc.org/Tutorial/PelicanTutorial.html

Contents

Introduction

PelicanHPC is a rapid (~5 minutes, when you know what you're doing) means of setting up a "high performance computing" (HPC) cluster for parallel computing using MPI. This tutorial gives a basic description of what PelicanHPC does, addresses how to use the released CD images to set up a HPC cluster, and gives some basic examples of usage.

Description of PelicanHPC

PelicanHPC is a distribution of GNU/Linux that runs as a "live CD" or bootable USB image (it can also be booted from a hard disk partition, or it can be used as a virtualized OS). If the ISO image file is put on a CD or USB, it can then be used to boot a computer. The computer on which PelicanHPC is booted is referred to as the "frontend node". It is the computer with which the user interacts. Once PelicanHPC is running, a script - "pelican_setup" - may be run. This script configures the frontend node as a netboot server. After this has been done, other computers can boot copies of PelicanHPC over the network. These other computers are referred to as "compute nodes". PelicanHPC configures the cluster made up of the frontend node and the compute nodes so that MPI-based parallel computing may be done.

A "live CD" such as PelicanHPC by default does not use the hard disks of any of the nodes (except Linux swap space, if it exists), so it will not destroy or alter your installed operating system. When the PelicanHPC cluster is shut down, all of the computers are in their original state, and will boot back into whatever operating systems are installed on them. PelicanHPC *can* optionally be made to use hard disk storage, so that its state can be preserved across boots. It can be configured to boot without user intervention, with access possible by ssh. There is also the possibility of making the compute nodes boot using wake-on-LAN. With these more advanced optional features, PelicanHPC can be used to run a headless permanent cluster.

PelicanHPC is made using Debian GNU/Linux as its base, through the Debian Live system. It is made by running a single script using the command "`sh make_pelican-v*`". Customized versions of PelicanHPC, for example, containing additional packages, can easily be made by modifying the make_pelican script. The make_pelican script and the needed packages are provided on PelicanHPC, so you can build a custom image using the provided images. You can also run make_pelican from any GNU/Linux distro if you install live-build and a few other packages.

Features

- The frontend node can be a real computer booted using a CD or a USB device, or a virtual machine that is booted using the CD image file. With this last option, PelicanHPC can be used at the same time as the normal work environment, which may be any of the common operating systems.
- The compute nodes are normally real computers, for maximum performance, but they can also be virtual.
- Supports MPI-based parallel computing using Fortran (77, 90), C, C++, GNU Octave and Python.
- Offers the Open MPI implementation of MPI.
- Cluster can be resized to add or remove nodes using the "pelican_restarthpc" command.
- Easily extensible to add packages. Also easily modifiable, since the PelicanHPC CD/USB image is created using a single script that relies on the Debian Live system. For this reason, the distributed version is fairly basic and lightweight.
- Contains example software: Linpack HPL (now at v2.0) benchmark and extensive examples that use GNU Octave. Also has mpi4py.

Limitations and requirements

- The compute nodes must be booted over the network. This is an option offered by all modern networking devices supplied with motherboards, but it often must be enabled in the BIOS setup. Enable it, and give it higher priority that booting from hard disk or other sources. If you have a network card that won't do netboot, it is possible to work around this using rom-o-matic. Another thing to be aware of is that the PelicanHPC frontend operates as a dhcp server. You should not use it on an open network, or you will cause dhcp conflicts. This will get you into a world of trouble with the network administrators. Plus, your compute nodes will not boot properly.
- A PelicanHPC cluster is designed to be used by a single person - there is only one user, with the username "user".
- Released versions are for 64 bit CPUs only (Opteron, Turion, Core 2, etc.). make_pelican can easily be used to make a 32 bit version, if needed.
- The PelicanHPC web page lists some other similar distros that may be more appropriate for certain uses.

Licensing and Disclaimer

PelicanHPC is a CD image made by running a script (see below). The script is licensed GPL v3. The resulting CD image contains software from the Debian distribution of GNU/Linux, and several other sources, which is subject to the licenses chosen by the authors of that software.

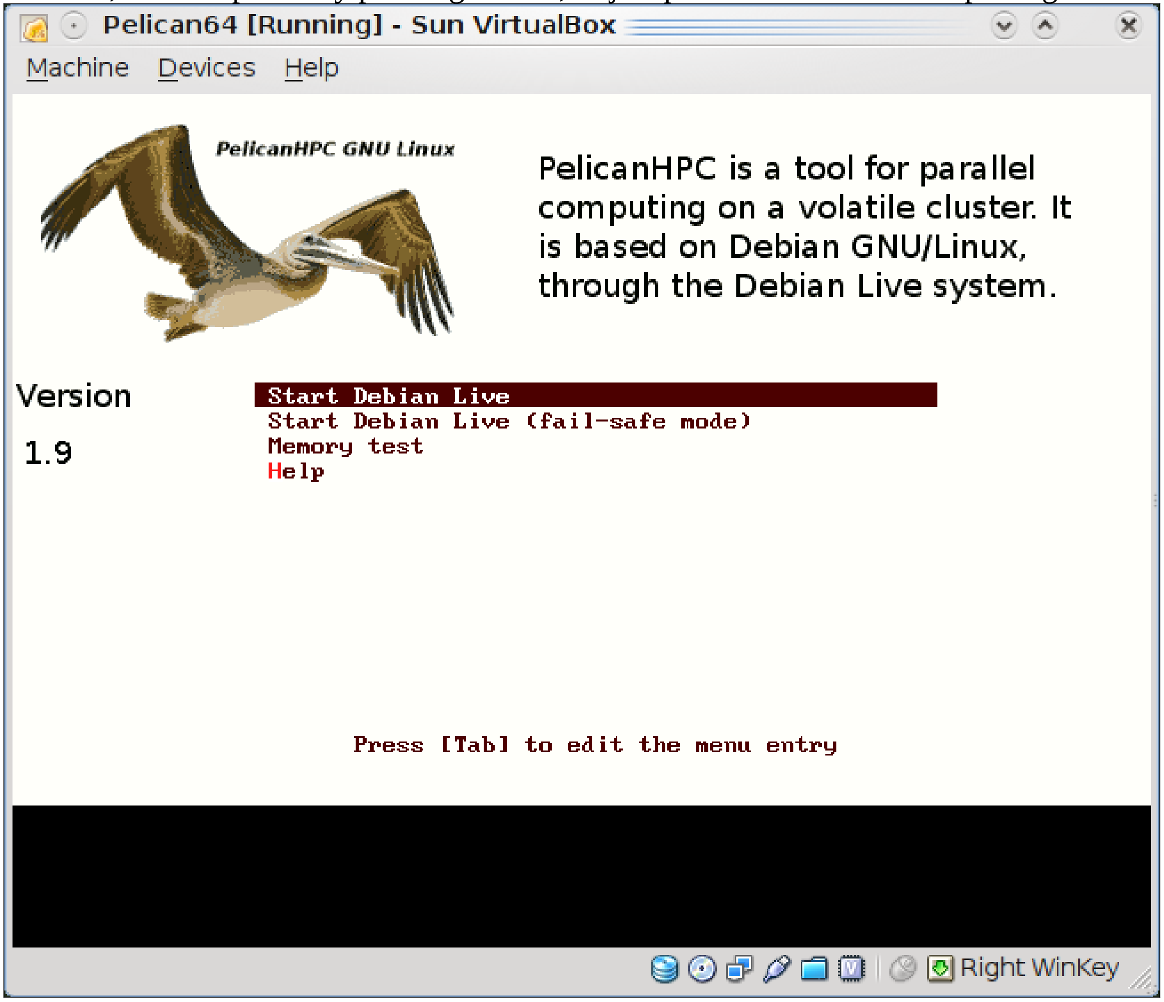
This released PelicanHPC CD images are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Initial setup

The two main commands for administration of the cluster are "pelican_setup", to configure the frontend as a server, NFS export /home, etc., and "pelican_restarthpc", which is used to add/remove nodes after the initial setup. The rest of this explains how this works.

The frontend and all compute nodes must be networked together. IMPORTANT: the frontend node will act as a DHCP server, so be sure to isolate the network used for the cluster from other networks, to avoid conflicts with other DHCP servers. If you start handing out IP addresses to your co-workers' computers, they may become annoyed. If the frontend node has multiple network interfaces, you can use one to connect to the cluster and another to connect to the Internet.

Put the CD in the computer that will be the frontend, and turn it on. Make sure the BIOS setup lets you boot from CD. When you boot up, you'll see something like the following. Here, if you press <Tab>, you have the opportunity to enter options to set keyboard mappings, or special "cheatcodes" to make the CD boot on problematic hardware. For example, I can get a Spanish keyboard by pressing <Tab> and then adding "keyb=es" to the default configuration. Either explore the options by highlighting the Help line and pressing <Enter>, or edit options by pressing <Tab>, or just press <Enter> to boot up using default settings.

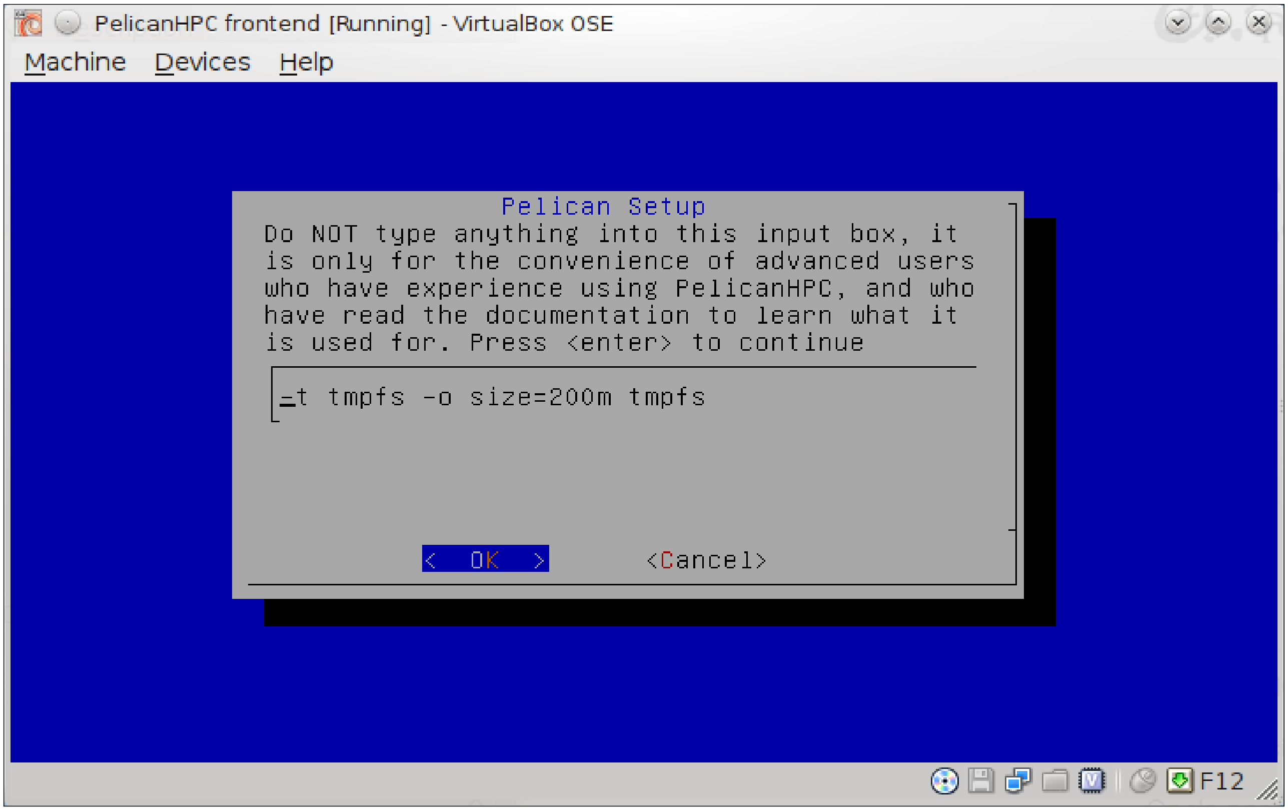


PelicanHPC is a tool for parallel computing on a volatile cluster. It is based on Debian GNU/Linux, through the Debian Live system.

Version 1.9

Press [Tab] to edit the menu entry

Once you boot up, eventually you see:

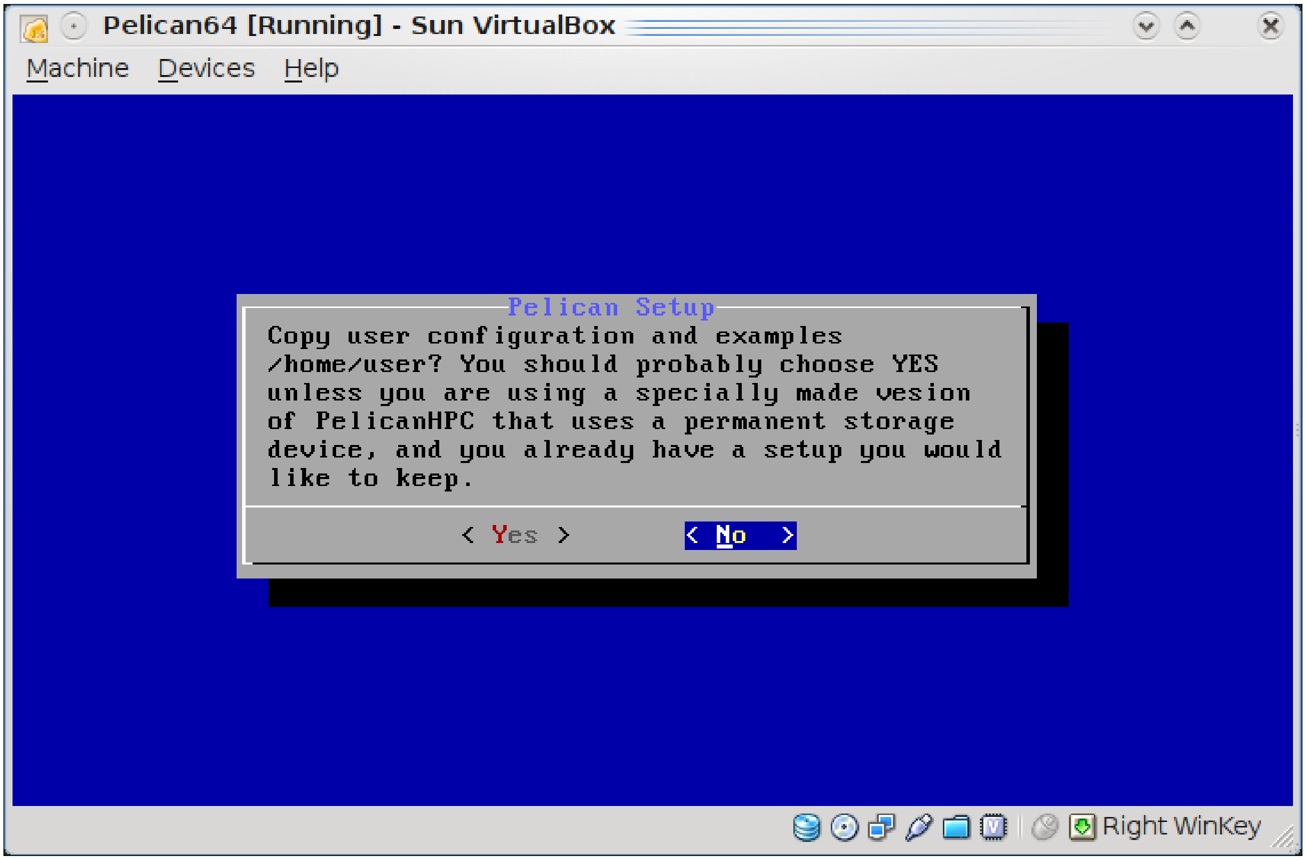


```
Pelican Setup
Do NOT type anything into this input box, it
is only for the convenience of advanced users
who have experience using PelicanHPC, and who
have read the documentation to learn what it
is used for. Press <enter> to continue

-t tmpfs -o size=200m tmpfs

< OK >        <Cancel>
```

This screen gives you the opportunity to use a permanent storage device for the /home directory of the PelicanHPC user. By default, if you just press <Enter>, hard disks are not used, and PelicanHPC does not alter any of the computers used in the cluster. This is safe and easy, but it has the disadvantage that any work you do disappears when you shut down the cluster. To use permanent storage, you can type in the name of a device (hard disk partition, USB drive, etc.) that has a formatted ext2 or ext3 partition, which will be mounted at /home. For example, you could replace "ram1" with "sda2" or "hdb5" (no quotes). If you do this, a directory "user" will be created at the root of the specified device, and will be used as the home directory of the cluster user (username "user"). If you shut down the cluster, the directory will not be removed, and it can be re-used when you restart PelicanHPC. If you have any doubts about this, just press <Enter>. For casual experimentation, you do not need this feature. This feature is provided as a convenience for advanced users. It is impossible to test this feature on all possible hardware configurations, so NO GUARANTEES ARE MADE THAT IT WILL NOT DESTROY YOUR HARD DISK. Back up your data before trying anything but the default.

IMPORTANT NOTE: there is another way to use permanent storage that is quite flexible. This is documented in the file /home/user/pelican_config, which you can see if you boot using the default. If this is your first experience with PelicanHPC, I recommend doing a default boot, study pelican_config, and then choose the option for permanent storage that you find most appropriate.

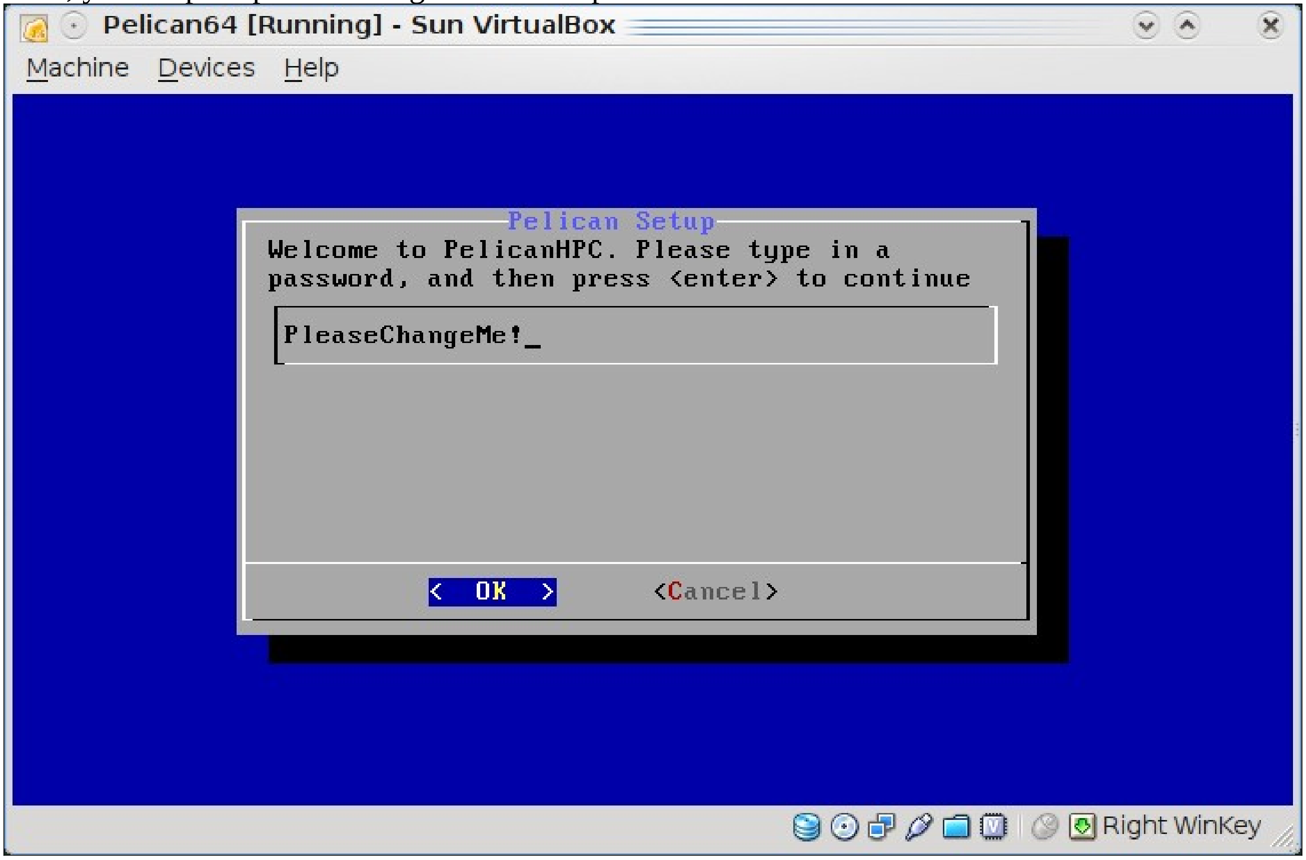
Next, you will see



```
Copy user configuration and examples
/home/user? You should probably choose YES
unless you are using a specially made vesion
of PelicanHPC that uses a permanent storage
device, and you already have a setup you would
like to keep.
```

You will probably want to choose "yes", unless you are re-using work you saved in a previous session.

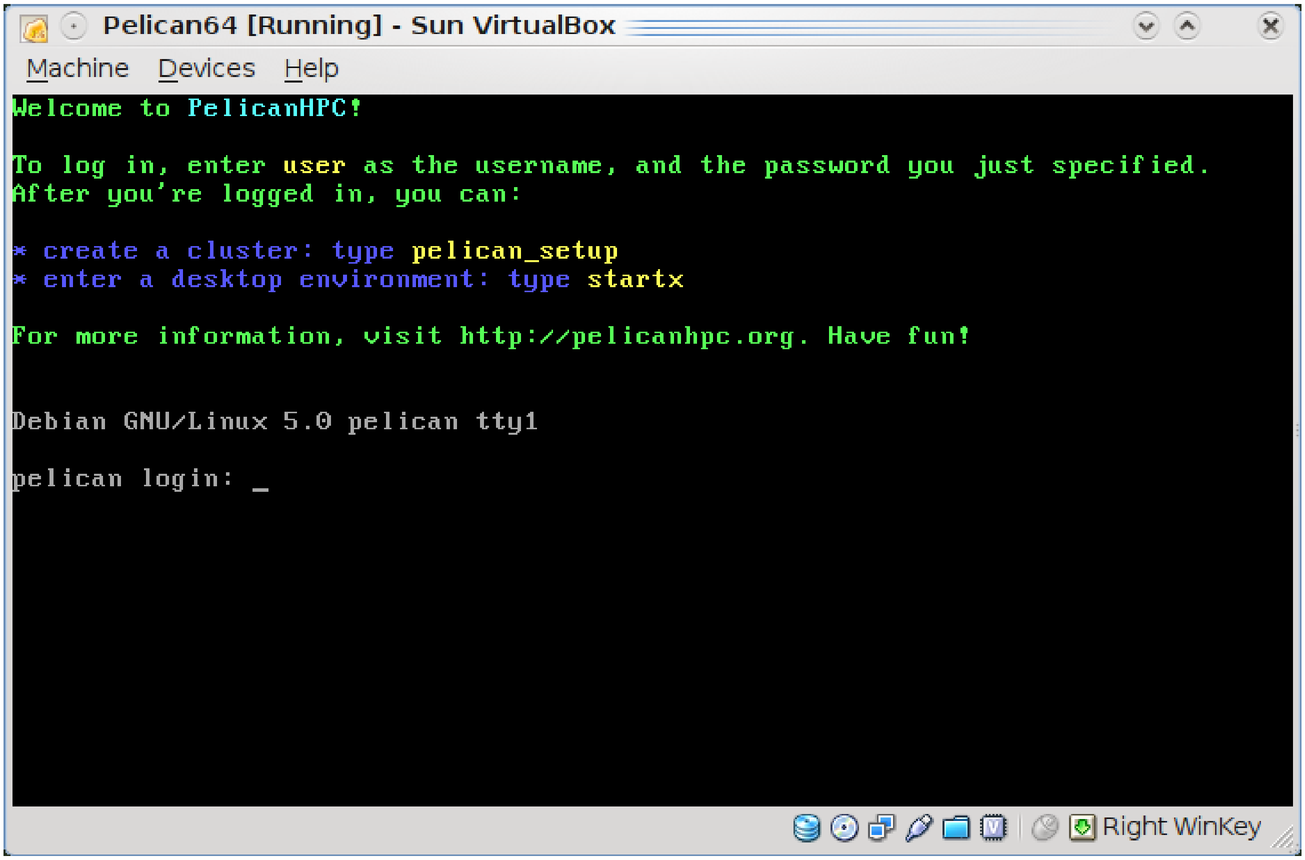
Next, you are prompted to change the default password:



```
Welcome to PelicanHPC. Please type in a
password, and then press <enter> to continue

PleaseChangeMe!_
```

You should backspace to remove the default and then type in a new password. This will be the password for user "user" on the frontend node and on all of the compute nodes, too.

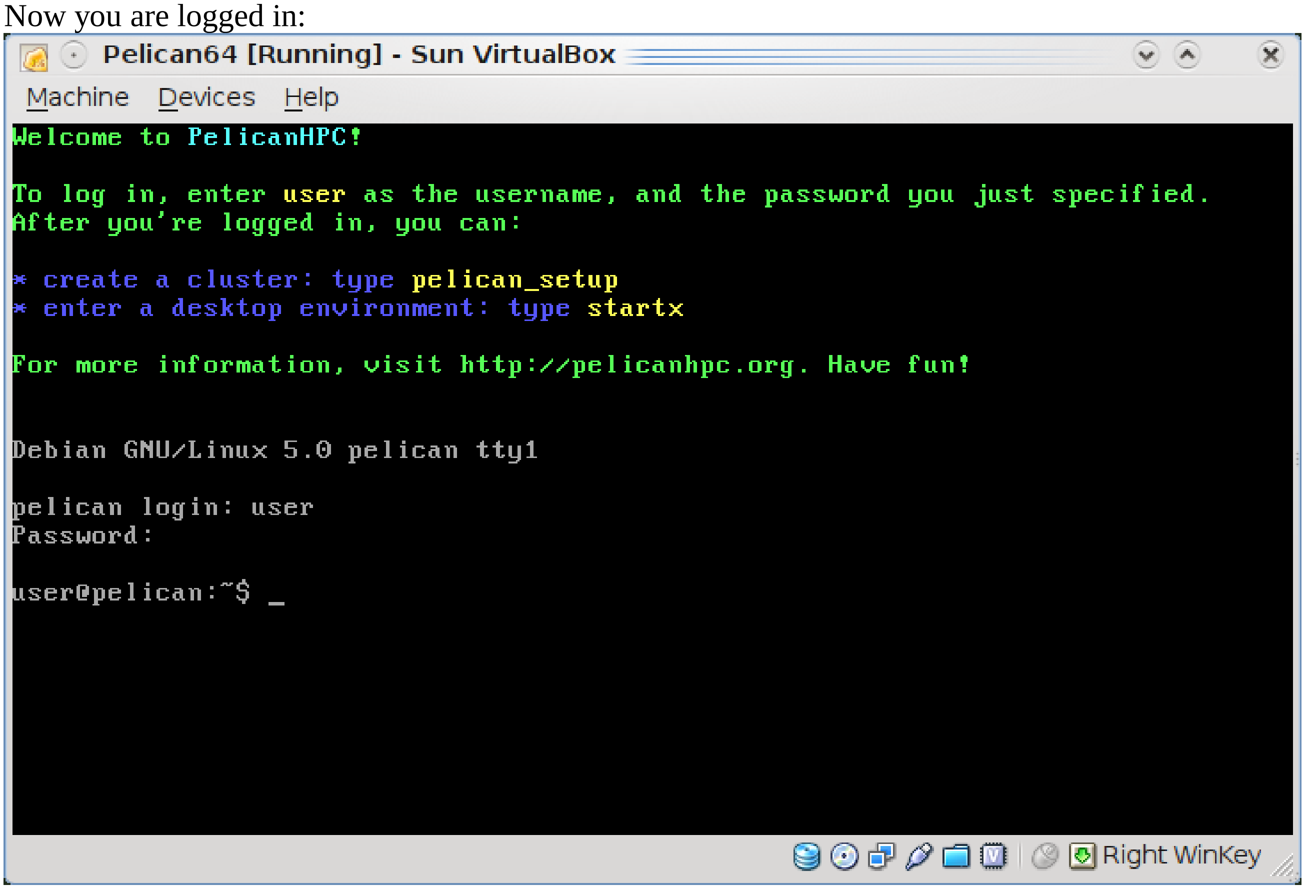
Finally, you are all booted up and the login prompt appears:



```
Welcome to PelicanHPC!

To log in, enter user as the username, and the password you just specified.
After you're logged in, you can:

* create a cluster: type pelican_setup
* enter a desktop environment: type startx

For more information, visit http://pelicanhpc.org. Have fun!


Debian GNU/Linux 5.0 pelican tty1

pelican login: _
```

Enter the username "user" and then the password that you set a moment ago.

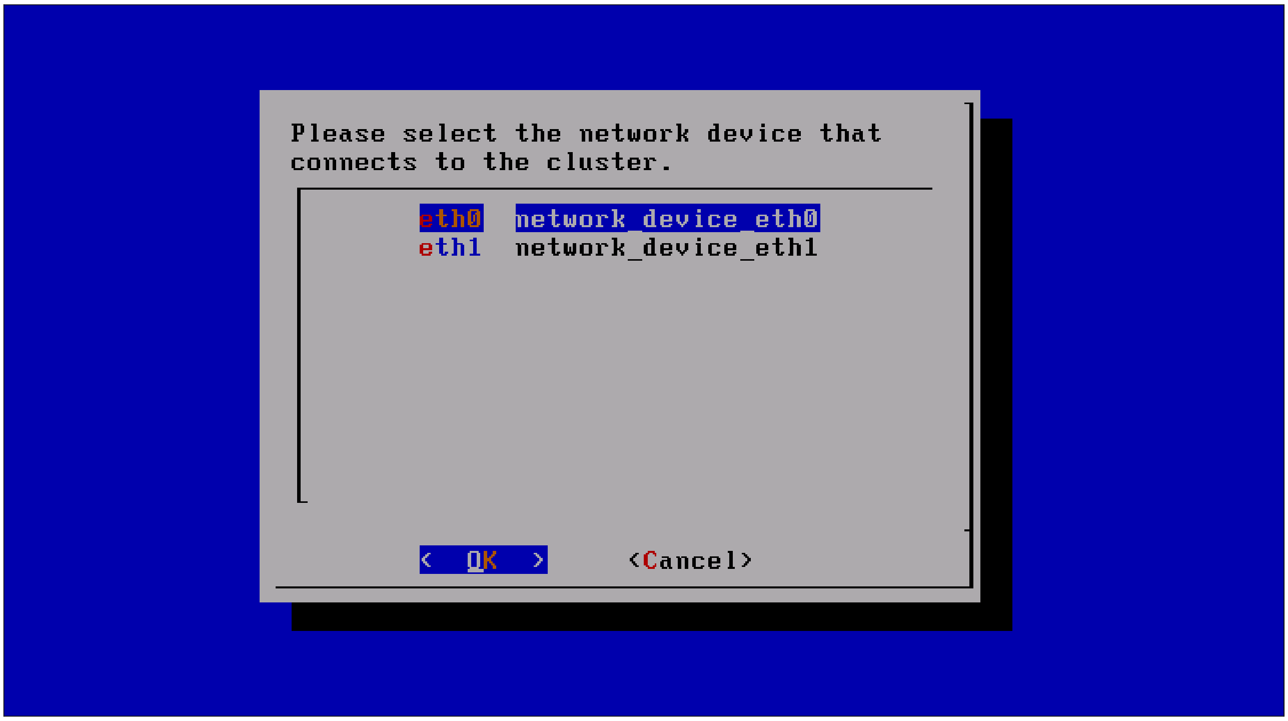
Now you are logged in:



```
Welcome to PelicanHPC!

To log in, enter user as the username, and the password you just specified.
After you're logged in, you can:

* create a cluster: type pelican_setup
* enter a desktop environment: type startx

For more information, visit http://pelicanhpc.org. Have fun!


Debian GNU/Linux 5.0 pelican tty1

pelican login: user
Password:

user@pelican:~$ _
```

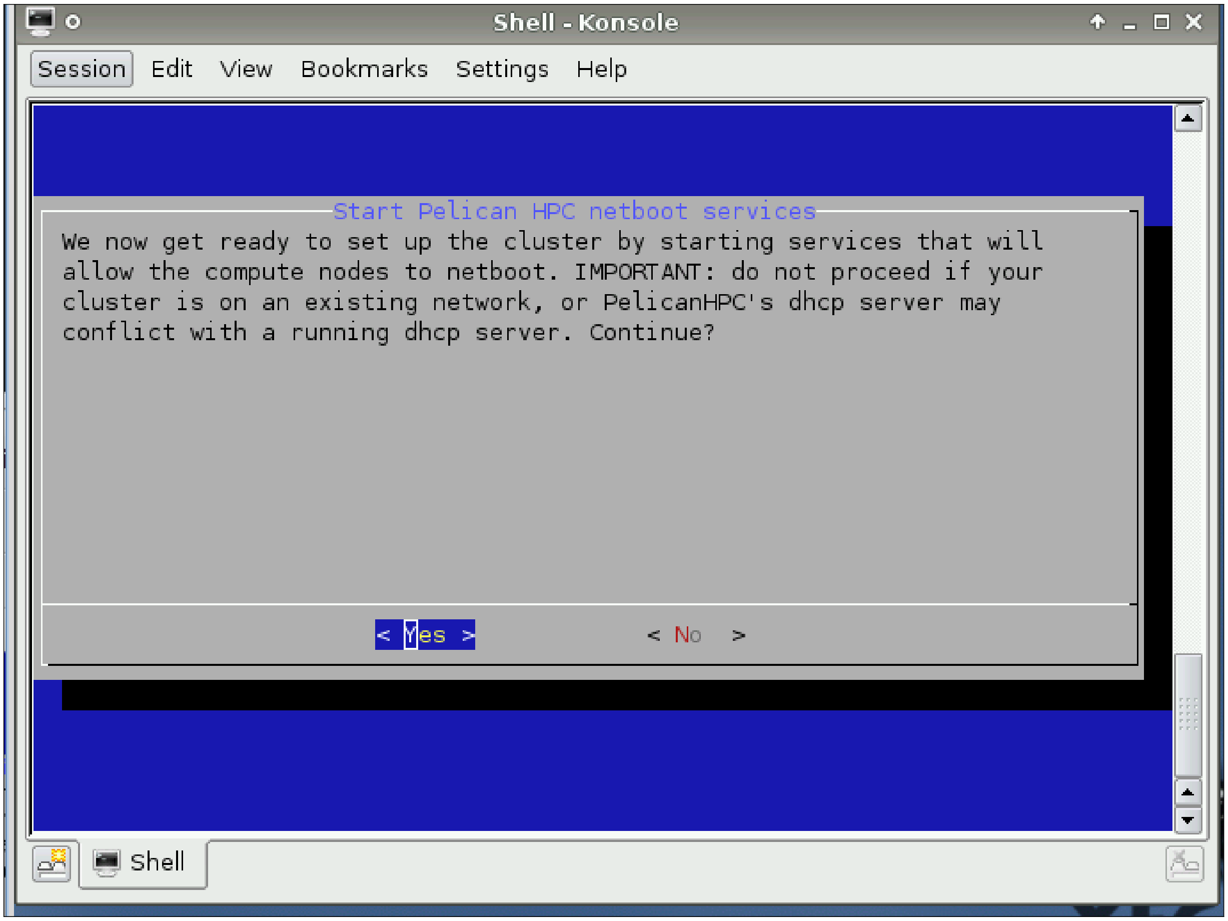
Note that you can enter the Xfce graphical enterface if you choose to by typing "startx". By default, PelicanHPC uses the console, just to avoid possible problems with unusual graphics hardware.

To set up a cluster, type "pelican_setup". You can do this from the console as in these instructions, or from Xfce by opening up a terminal. Next, we see the following, supposing that you have more than 1 network device:



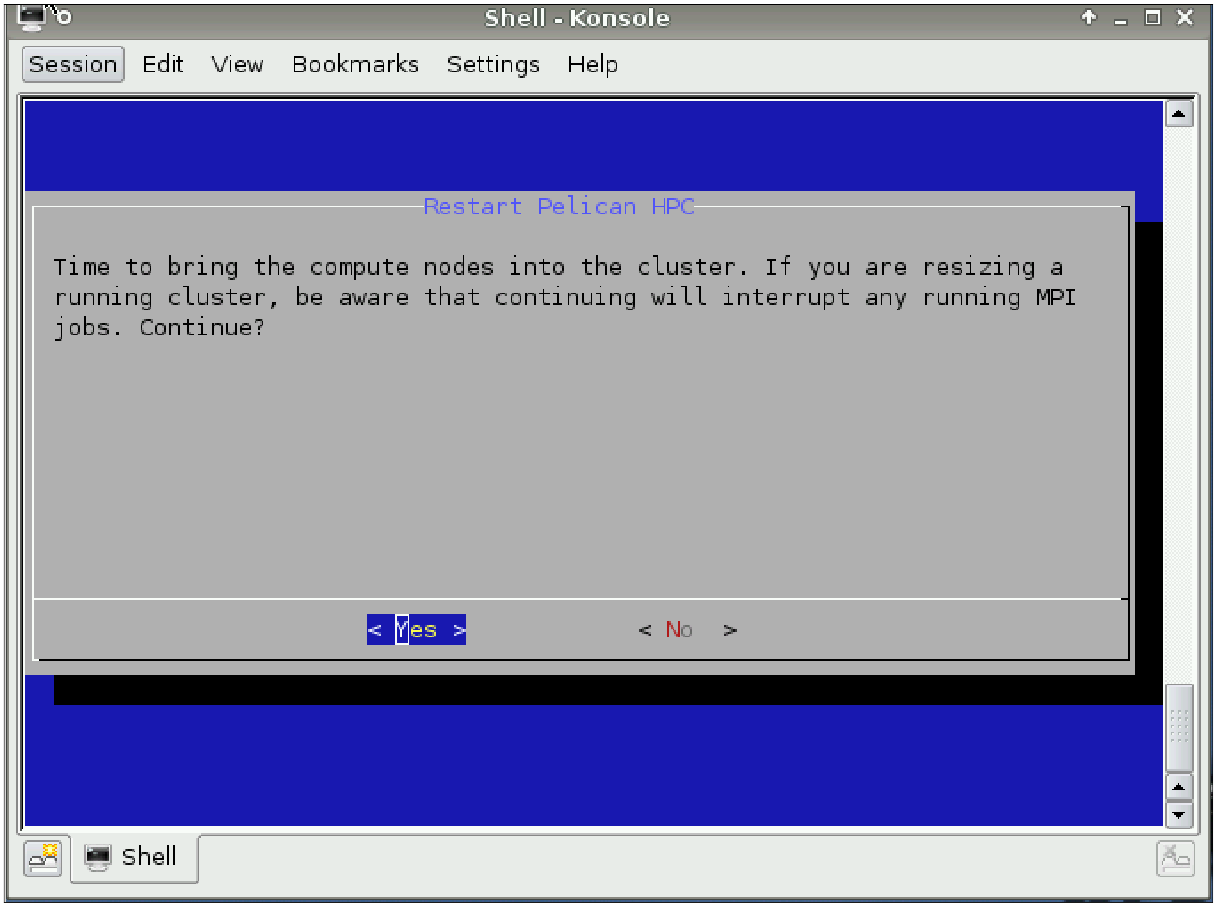
```
Please select the network device that
connects to the cluster.

        eth0   network_device_eth0
        eth1   network_device_eth1

        <  OK  >       <Cancel>
```

After you choose the net device, services need to be started. Please read the warning in the following screenshot. Setting up a PelicanHPC dhcp server will get you in trouble with your network administrators if you do this on an open network. You should make sure that the network device used for the cluster is isolated from all networks except the cluster. When you see the following screen, choose "yes".



```
Start Pelican HPC netboot services
We now get ready to set up the cluster by starting services that will
allow the compute nodes to netboot. IMPORTANT: do not proceed if your
cluster is on an existing network, or PelicanHPC's dhcp server may
conflict with a running dhcp server. Continue?

        < Yes >          < No  >
```

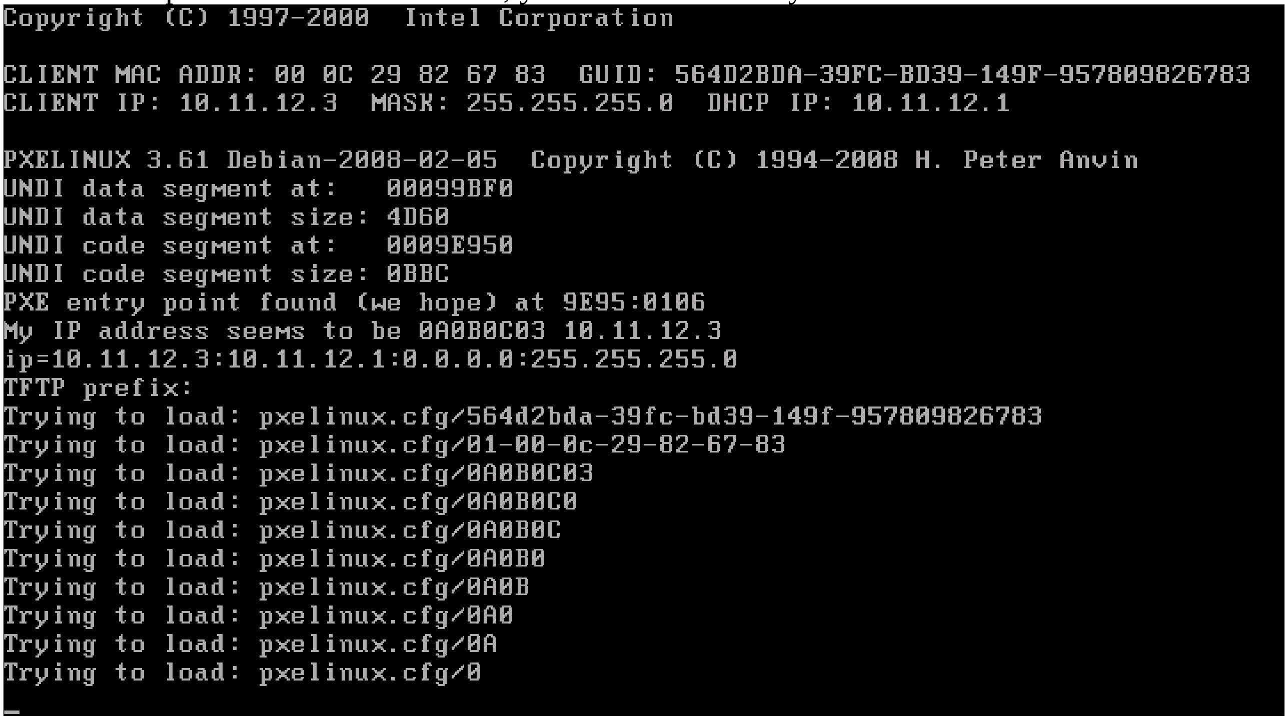
Next you will see



```
Restart Pelican HPC

Time to bring the compute nodes into the cluster. If you are resizing a
running cluster, be aware that continuing will interrupt any running MPI
jobs. Continue?

        < Yes >          < No  >
```

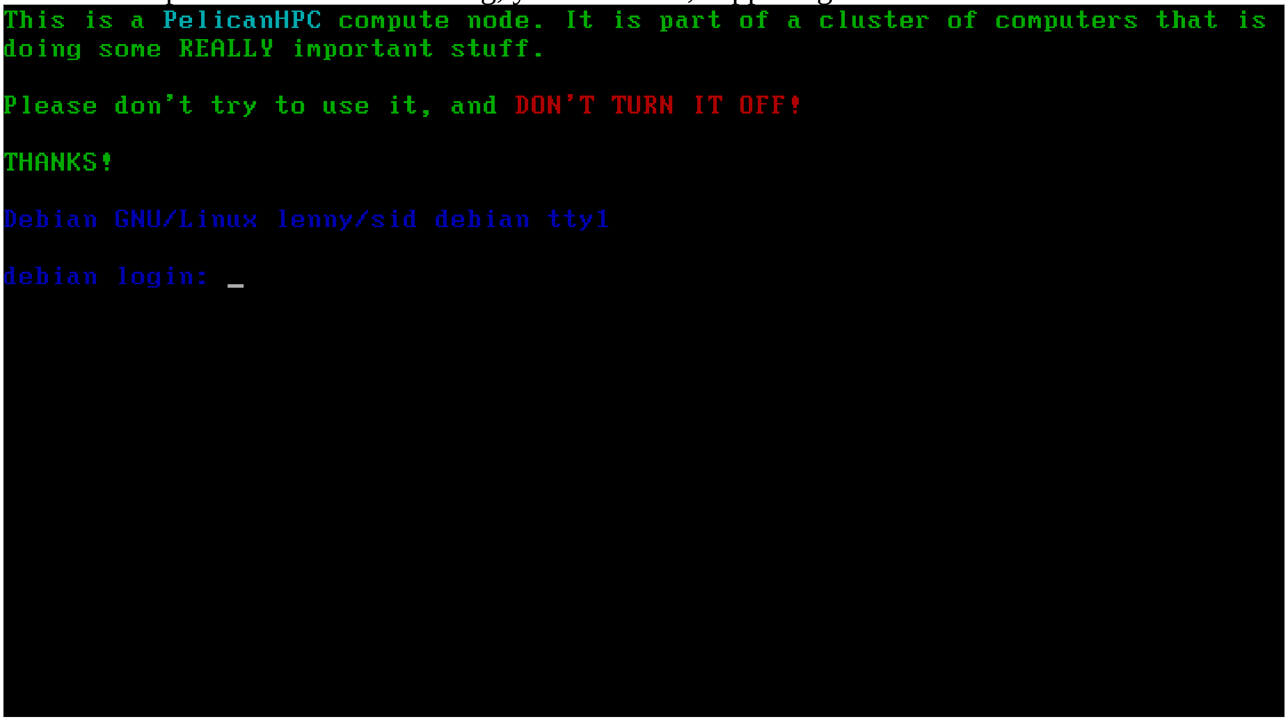
Press enter, and go turn on the compute nodes.

When a compute node starts to netboot, you'll see this whiz by:



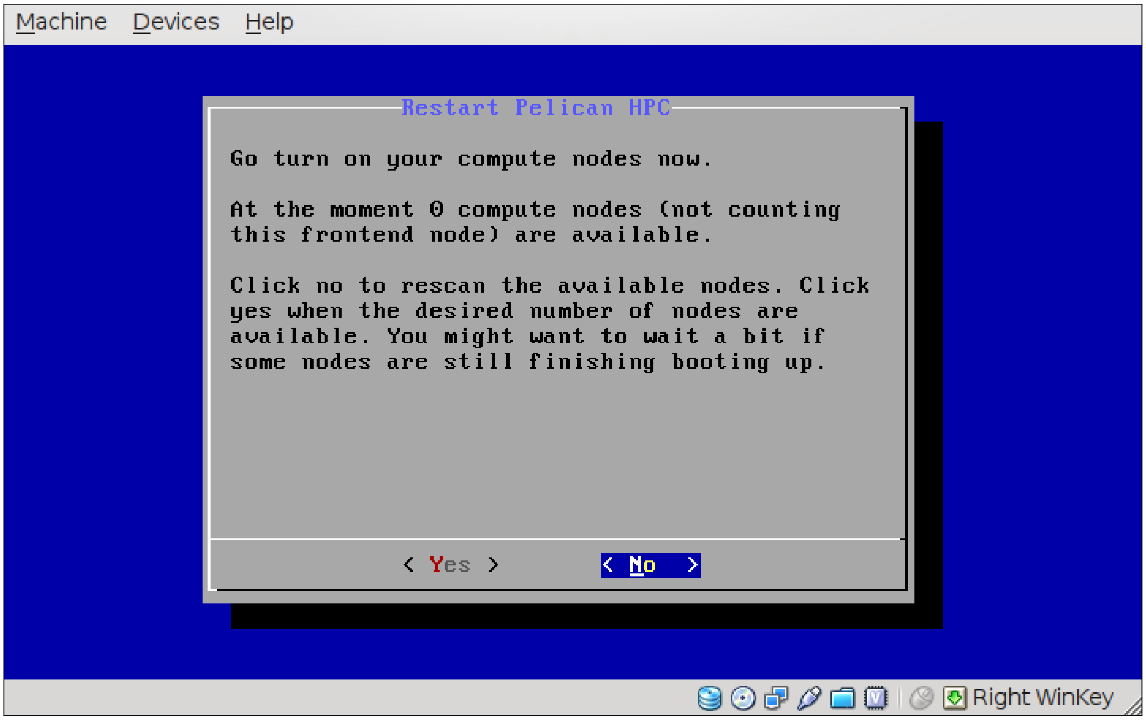
```
Copyright (C) 1997-2000  Intel Corporation

CLIENT MAC ADDR: 00 0C 29 82 67 83  GUID: 564D2BDA-39FC-BD39-149F-957809826783
CLIENT IP: 10.11.12.3  MASK: 255.255.255.0  DHCP IP: 10.11.12.1

PXELINUX 3.61 Debian-2008-02-05  Copyright (C) 1994-2008 H. Peter Anvin
UNDI data segment at:   00099BF0
UNDI data segment size: 4D60
UNDI code segment at:   0009E950
UNDI code segment size: 0BBC
PXE entry point found (we hope) at 9E95:0106
My IP address seems to be 0A0B0C03 10.11.12.3
ip=10.11.12.3:10.11.12.1:0.0.0.0:255.255.255.0
TFTP prefix:
Trying to load: pxelinux.cfg/564d2bda-39fc-bd39-149f-957809826783
Trying to load: pxelinux.cfg/01-00-0c-29-82-67-83
Trying to load: pxelinux.cfg/0A0B0C03
Trying to load: pxelinux.cfg/0A0B0C0
Trying to load: pxelinux.cfg/0A0B0C
Trying to load: pxelinux.cfg/0A0B0
Trying to load: pxelinux.cfg/0A0B
Trying to load: pxelinux.cfg/0A0
Trying to load: pxelinux.cfg/0A
Trying to load: pxelinux.cfg/0
_
```

When a compute node is done booting, you'll see this, supposing that it has a monitor:



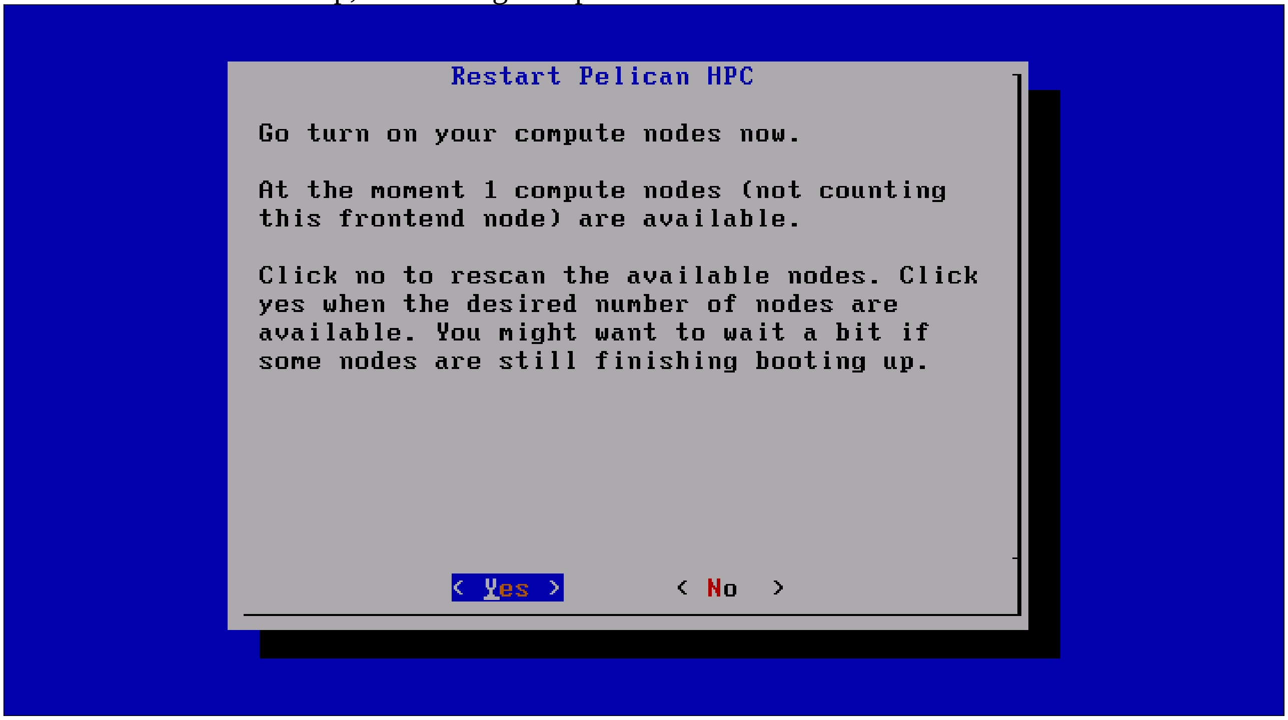
```
This is a PelicanHPC compute node. It is part of a cluster of computers that is
doing some REALLY important stuff.

Please don't try to use it, and DON'T TURN IT OFF!

THANKS!

Debian GNU/Linux lenny/sid debian tty1

debian login: _
```

Back on the frontend node, you see the following:



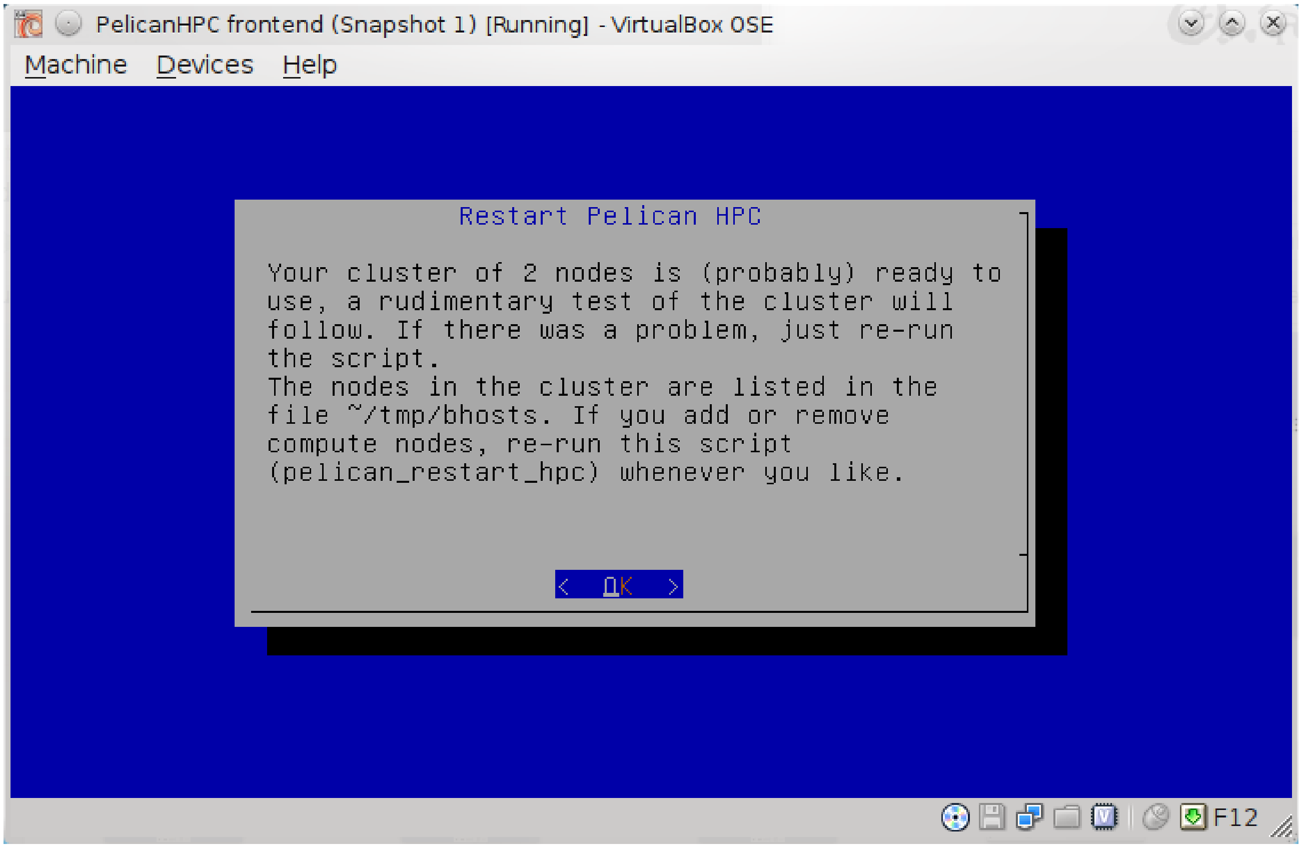
```
Restart Pelican HPC

Go turn on your compute nodes now.

At the moment 0 compute nodes (not counting
this frontend node) are available.

Click no to rescan the available nodes. Click
yes when the desired number of nodes are
available. You might want to wait a bit if
some nodes are still finishing booting up.

        < Yes >          < No >
```

Once a node has booted up, the count goes up:



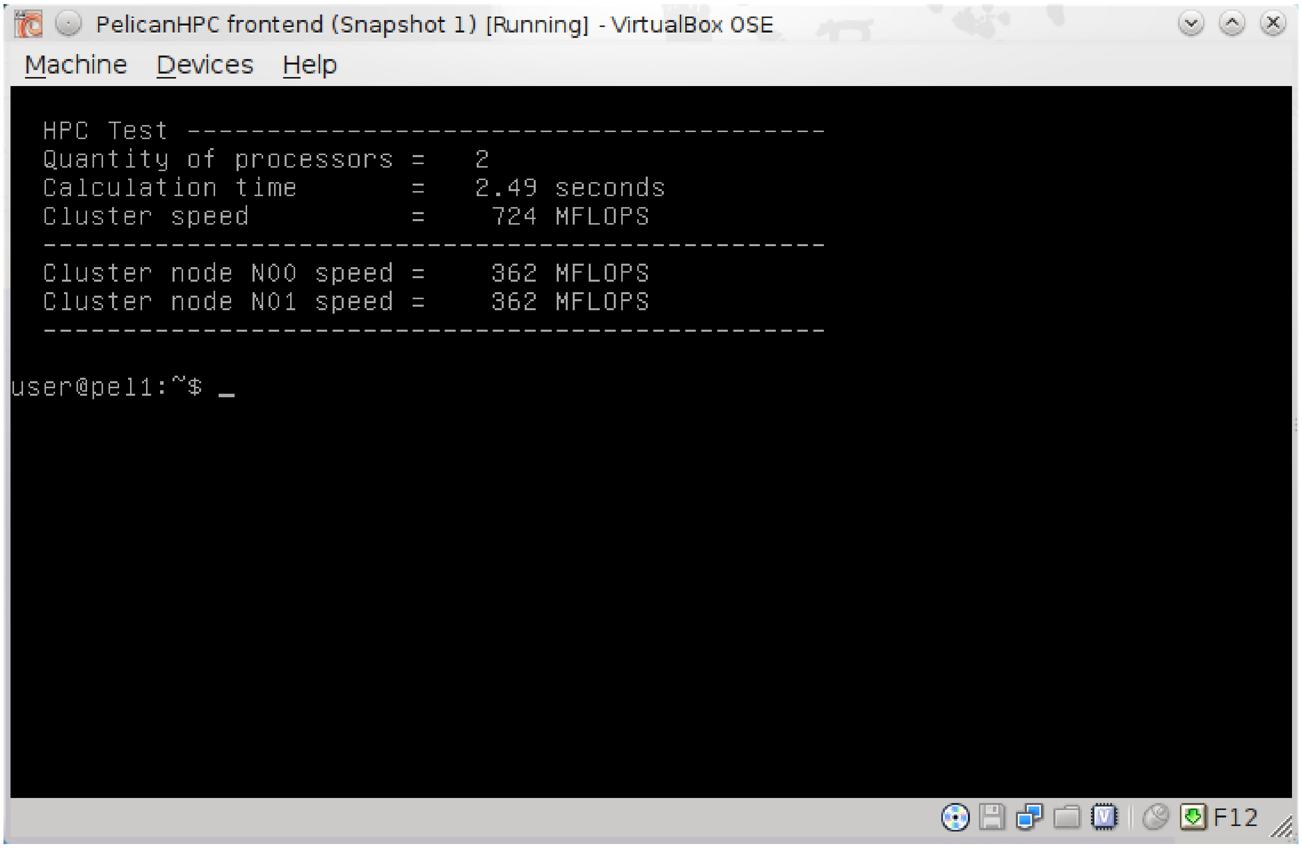
```
Restart Pelican HPC

Go turn on your compute nodes now.

At the moment 1 compute nodes (not counting
this frontend node) are available.

Click no to rescan the available nodes. Click
yes when the desired number of nodes are
available. You might want to wait a bit if
some nodes are still finishing booting up.

        < Yes >          < No >
```

Keep hitting "no" until all of your compute nodes have booted up. Once you click yes, you'll see something like the following, depending on how many nodes you have.



```
Restart Pelican HPC

Your cluster of 2 nodes is (probably) ready to
use, a rudimentary test of the cluster will
follow. If there was a problem, just re-run
the script.
The nodes in the cluster are listed in the
file ~/tmp/bhosts. If you add or remove
compute nodes, re-run this script
(pelican_restart_hpc) whenever you like.

                    < OK >
```

Finally, a quick test of the cluster is run. You should see something like the following:



```
  HPC Test --------------------------------------------
  Quantity of processors =    2
  Calculation time       =    2.49 seconds
  Cluster speed          =     724 MFLOPS
  -----------------------------------------------------
  Cluster node N00 speed =     362 MFLOPS
  Cluster node N01 speed =     362 MFLOPS
  -----------------------------------------------------

user@pel1:~$ _
```

OK, that's it, the cluster is ready to use. Some other tips:

- you can add software to the frontend node using "apt-get install whatever", supposing that the

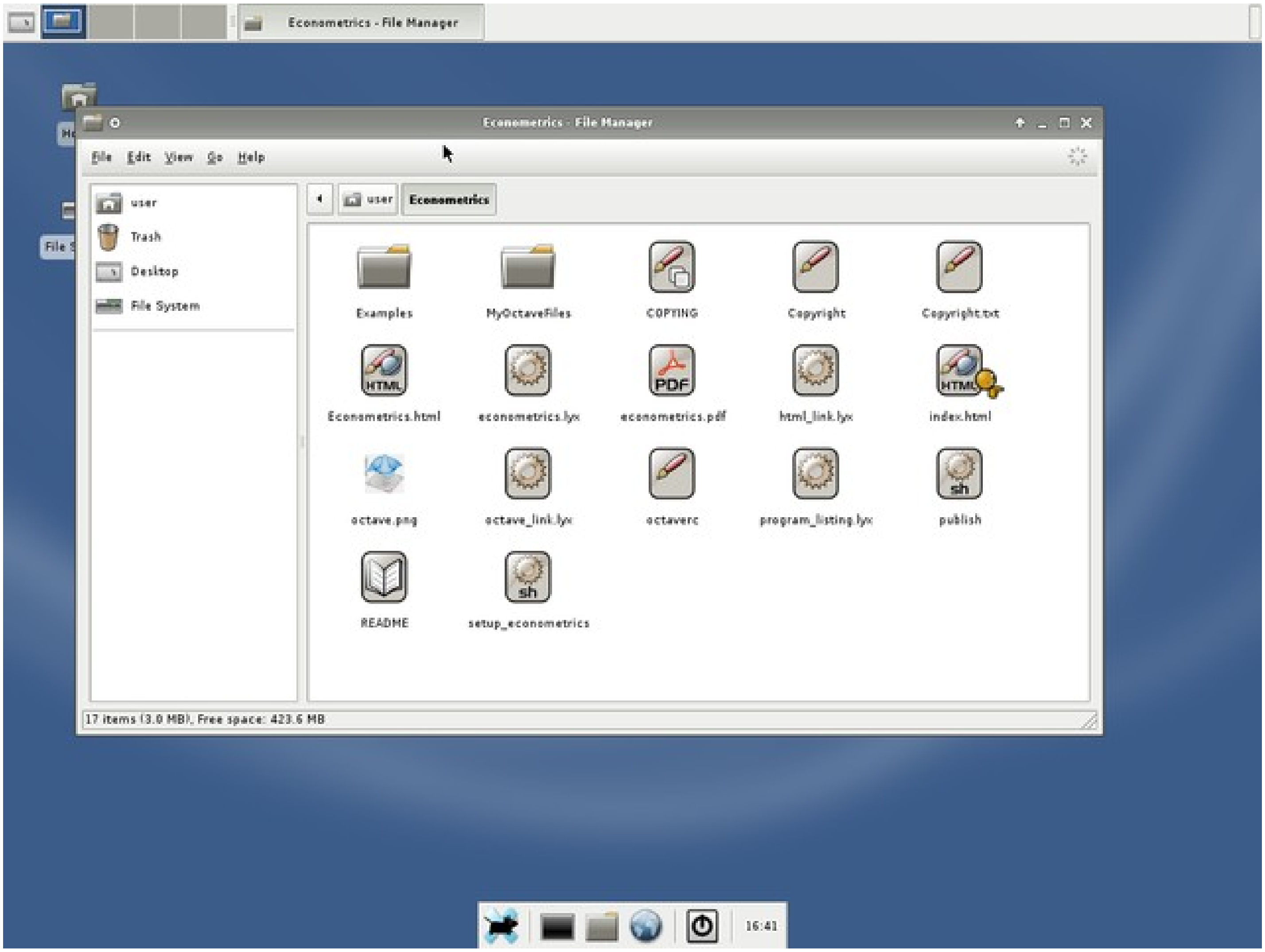
frontend has a second net card that you have configured to enable Internet access. This software is not available on the compute nodes. To add software so that it is available to all the nodes, it should be installed somewhere in /home/user.
- the default MPI setup is in the file /home/user/tmp/bhosts. This assigns ranks to hosts in a round robin fashion. If your hosts have different speeds, numbers of cores, etc., you should modify this file. If the frontend node is virtual but the compute nodes are real, you should probably remove the frontend node from the calculations.
- ksysguard is available, and a small amount of effort will turn it into a nice cluster monitor. See this post for general information on how to do it.
- if you need other packages, then you can make your own version pretty easily using the make_pelican script that is available on the PelicanHPC homepage. This is explained (somewhat) below.
- You can resize the cluster (add or remove compute nodes) whenever you like, by running "pelican_restarthpc".

IMPORTANT: In the /home/user directory is the file pelican_config. This file contains switches for advanced options that allow features such as use of permanent storage, booting without user intervention, changing the network of the cluster, wake-on-LAN, etc. Casual users do not need to explore this, but people who want a permanent cluster should look at it. It is self-documented.

Return to contents

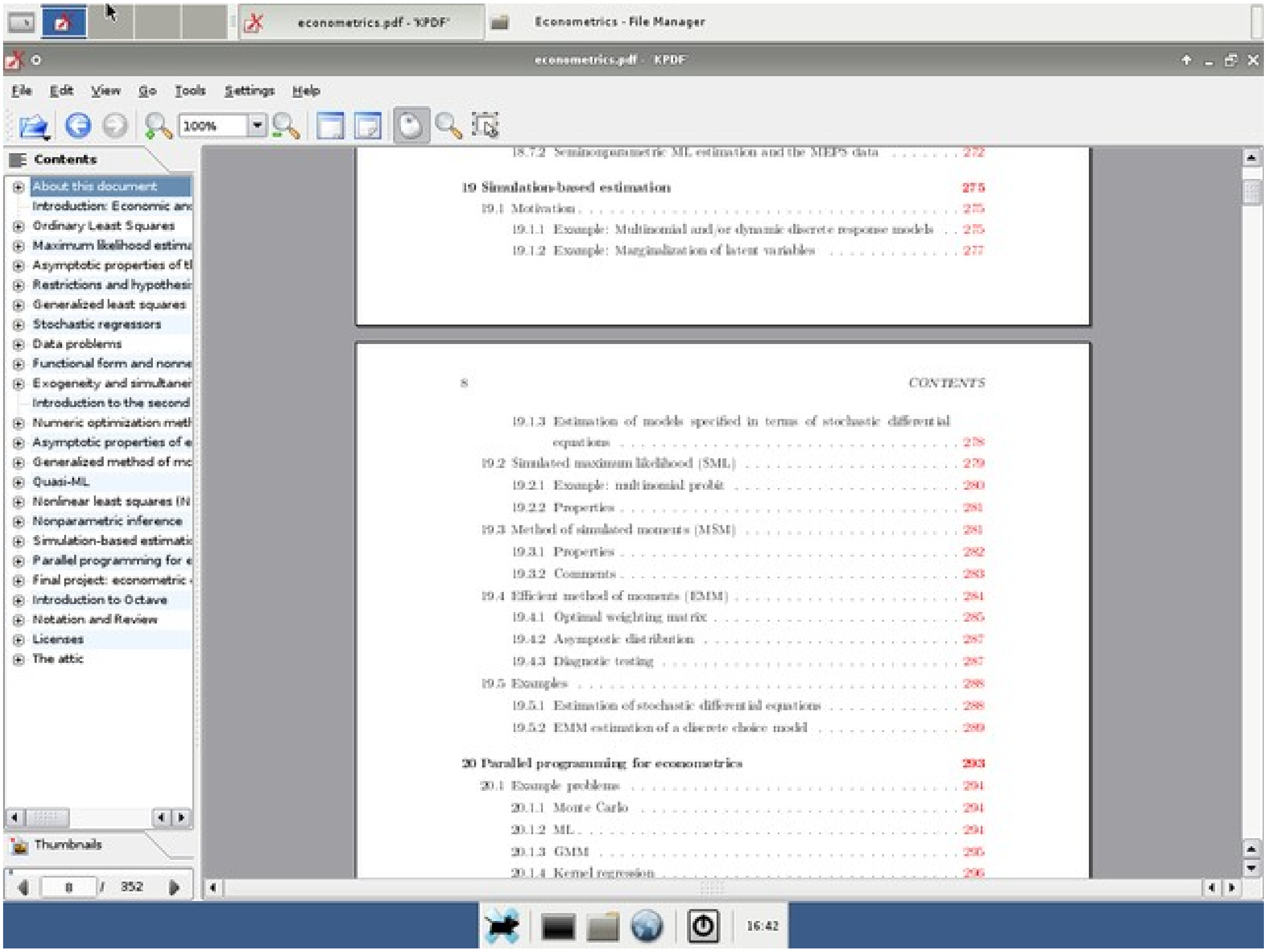
Example software

PelicanHPC has the Linpack HPL benchmark, and some extensive examples from the field of econometrics that use GNU Octave. Econometrics is a field of study that applies statistical methods to economic models. The software is in the Econometrics directory:



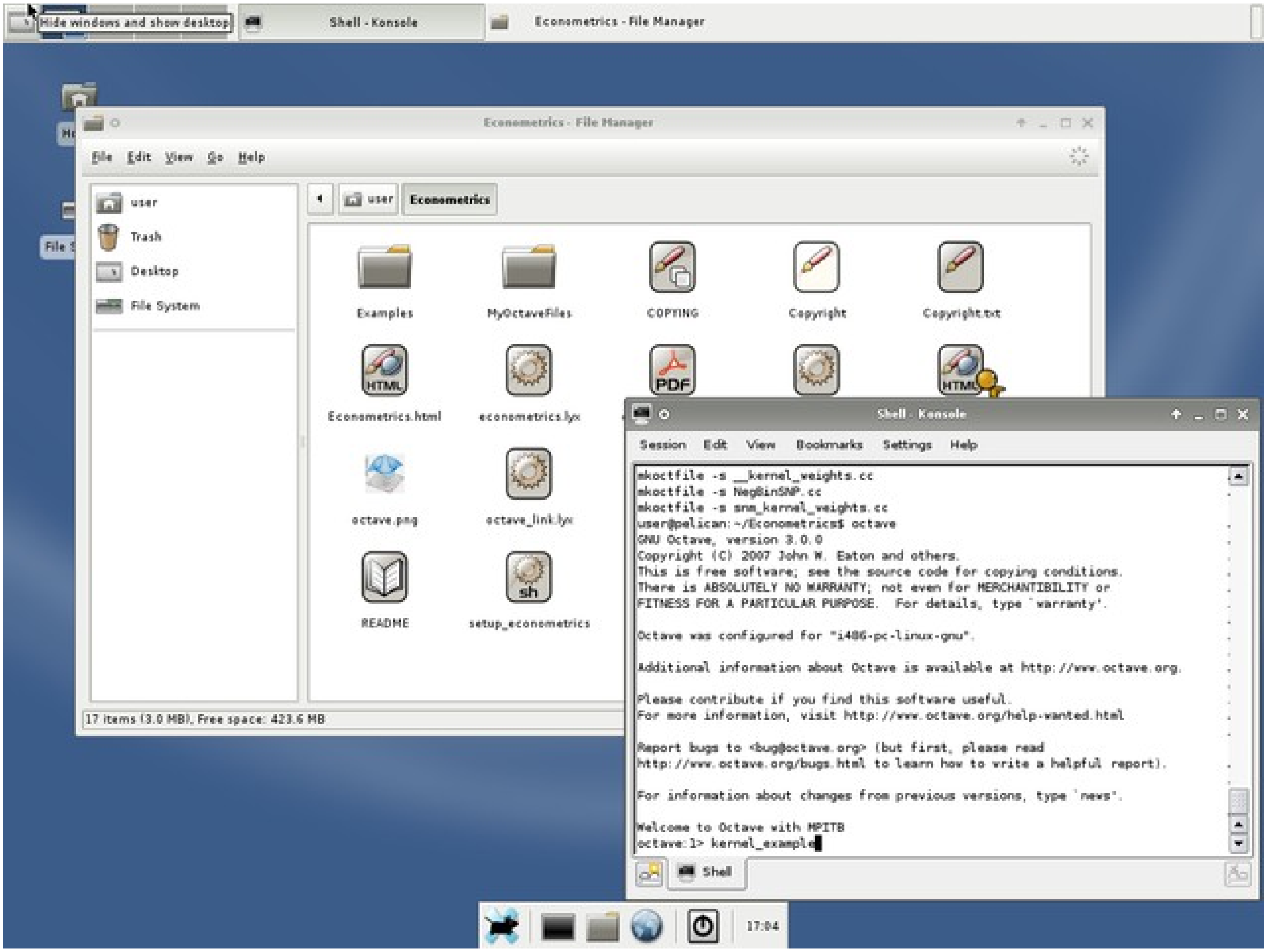
17 items (3.0 MB), Free space: 423.6 MB

There is a document "econometrics.pdf" that has a lot of information, including some about parallel computing:



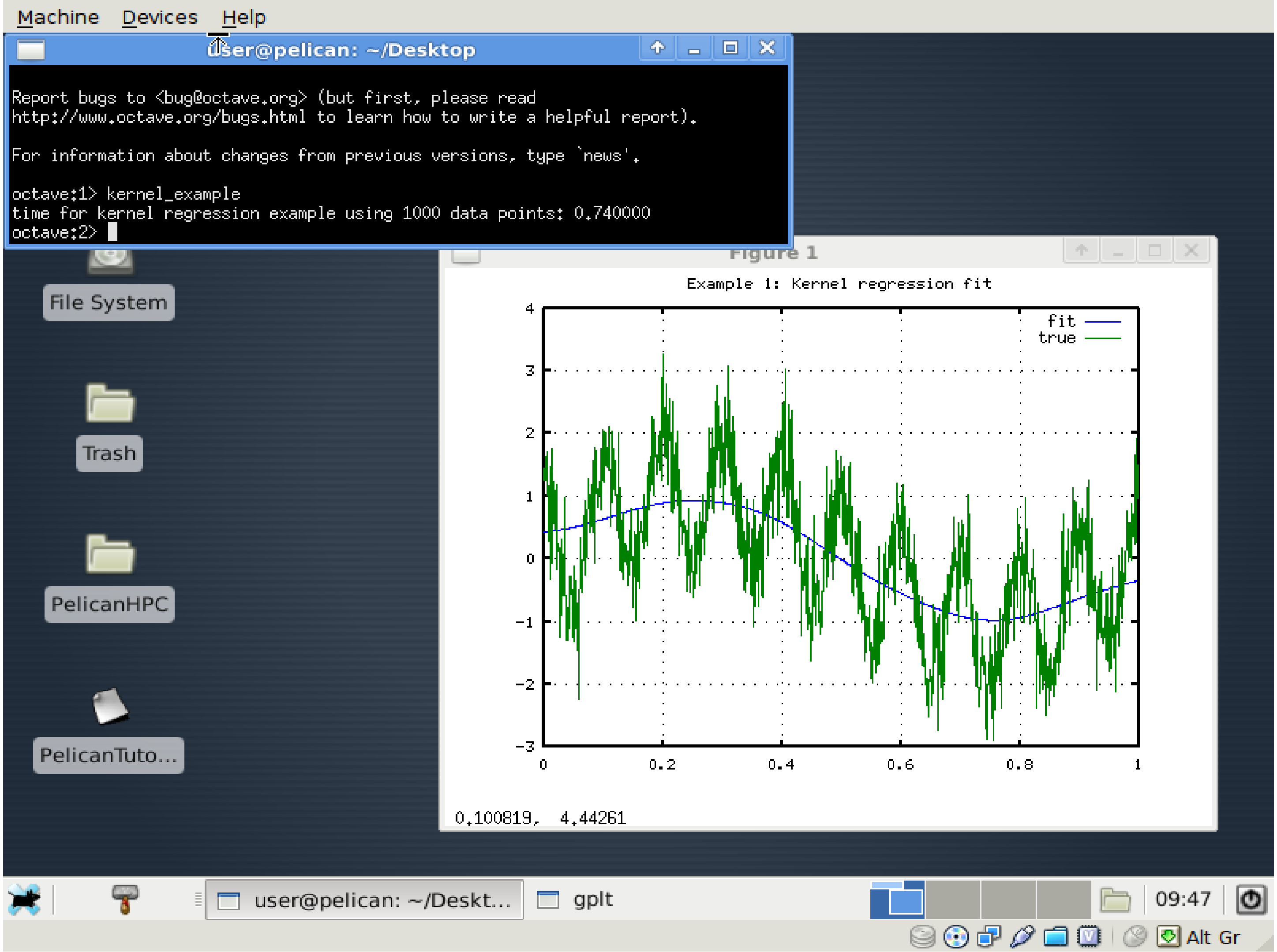
8 *CONTENTS*

Open a terminal, type "octave" and then "kernel_example" (please note that underscore back there, ... music please, maestro):



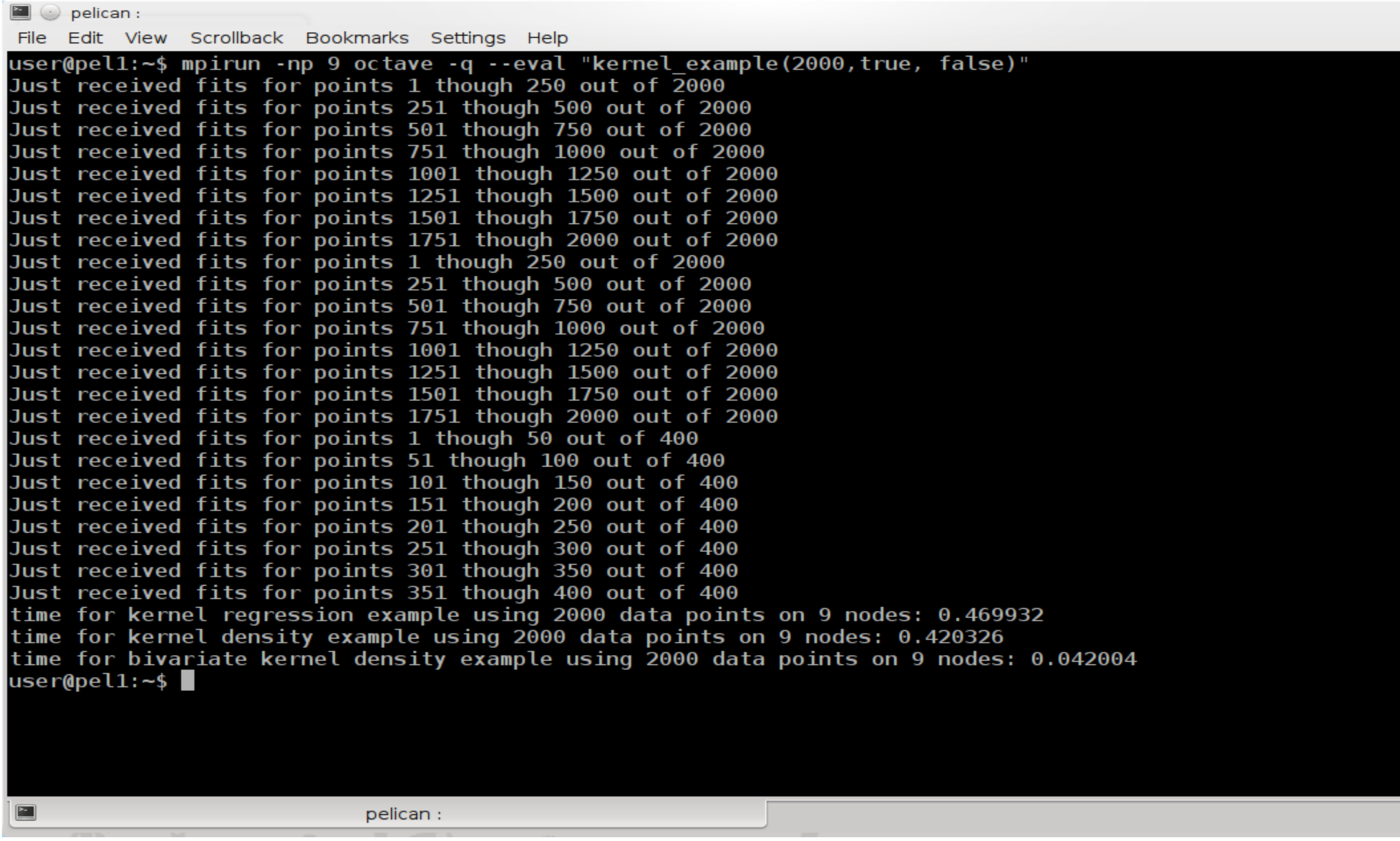
```
mkoctfile -s __kernel_weights.cc
mkoctfile -s NegBinSNP.cc
mkoctfile -s snm_kernel_weights.cc
user@pelican:~/Econometrics$ octave
GNU Octave, version 3.0.0
Copyright (C) 2007 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTIBILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type `warranty'.

Octave was configured for "i486-pc-linux-gnu".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html

Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).

For information about changes from previous versions, type `news'.

Welcome to Octave with MPITB
octave:1> kernel_example
```

et viola! some nice pictures:



```
Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).

For information about changes from previous versions, type `news'.

octave:1> kernel_example
time for kernel regression example using 1000 data points: 0.740000
octave:2> 
```

That last picture screenshot shows the output of kernel_example.m if it is run serially, on a single core. To see how to run it in parallel, see the next shot. **NOTE: the kernel routines do no computations on rank 0 (it is used to gather the results), so you must specify at least 2 MPI ranks.**



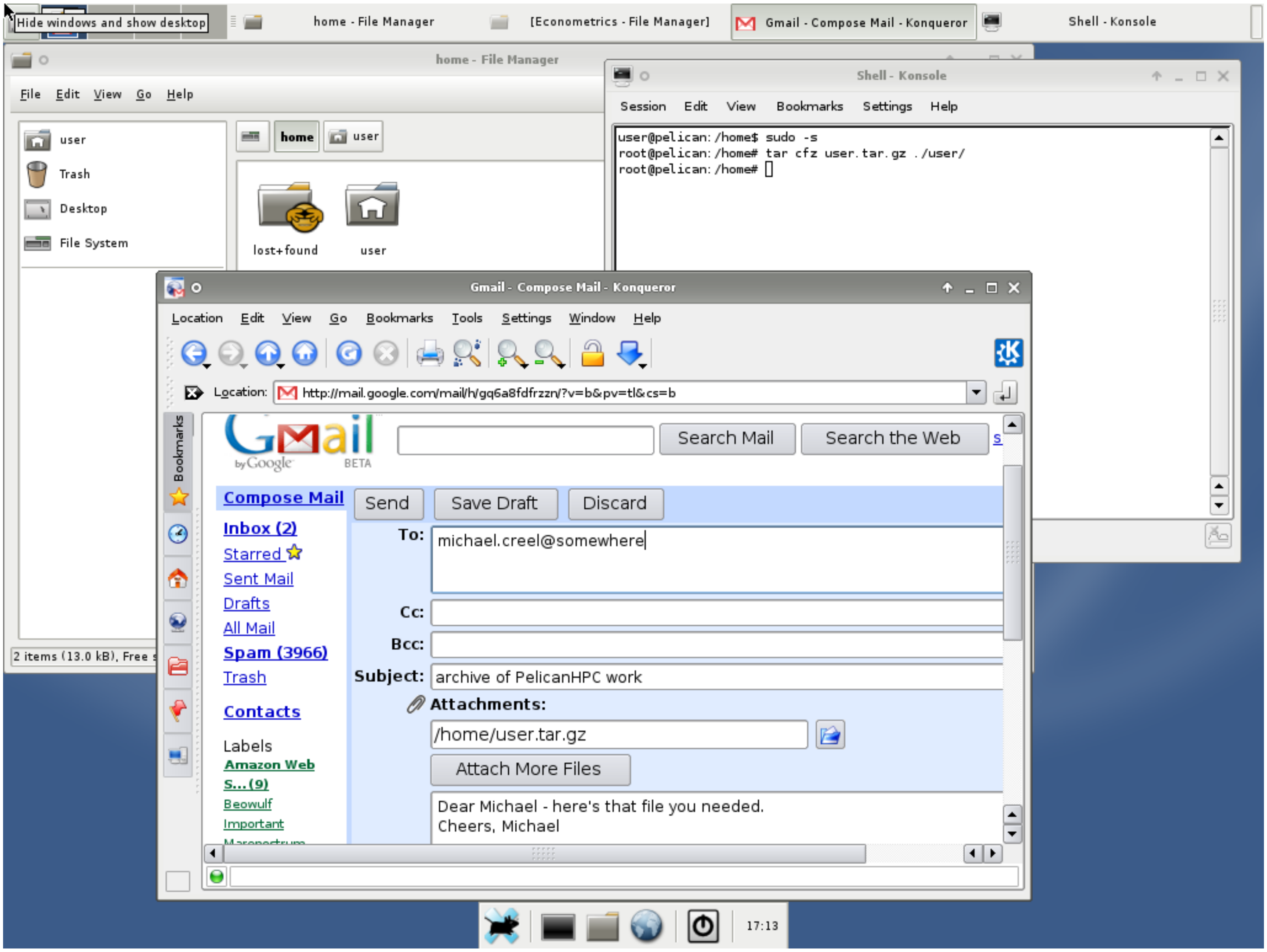
```
user@pel1:~$ mpirun -np 9 octave -q --eval "kernel_example(2000,true, false)"
Just received fits for points 1 though 250 out of 2000
Just received fits for points 251 though 500 out of 2000
Just received fits for points 501 though 750 out of 2000
Just received fits for points 751 though 1000 out of 2000
Just received fits for points 1001 though 1250 out of 2000
Just received fits for points 1251 though 1500 out of 2000
Just received fits for points 1501 though 1750 out of 2000
Just received fits for points 1751 though 2000 out of 2000
Just received fits for points 1 though 250 out of 2000
Just received fits for points 251 though 500 out of 2000
Just received fits for points 501 though 750 out of 2000
Just received fits for points 751 though 1000 out of 2000
Just received fits for points 1001 though 1250 out of 2000
Just received fits for points 1251 though 1500 out of 2000
Just received fits for points 1501 though 1750 out of 2000
Just received fits for points 1751 though 2000 out of 2000
Just received fits for points 1 though 50 out of 400
Just received fits for points 51 though 100 out of 400
Just received fits for points 101 though 150 out of 400
Just received fits for points 151 though 200 out of 400
Just received fits for points 201 though 250 out of 400
Just received fits for points 251 though 300 out of 400
Just received fits for points 301 though 350 out of 400
Just received fits for points 351 though 400 out of 400
time for kernel regression example using 2000 data points on 9 nodes: 0.469932
time for kernel density example using 2000 data points on 9 nodes: 0.420326
time for bivariate kernel density example using 2000 data points on 9 nodes: 0.042004
user@pel1:~$ 
```

Other things to try are "bfgsmin_example", "mle_example", "gmm_example", "mc_example" and a few others I'm forgetting about. To find where the code is, type "help mc_example", for example, while in Octave. Then, go edit the relevant file to learn more about what it does. Or, while in Octave, type "edit bfgsmin_example" (or edit whatever you like) and the file found in Octave's path will open up in the vim editor.

Return to contents

Saving your work

By default, PelicanHPC images put /home/user on a ramdisk which disappears when you shut down. You need to save your work between sessions, if you want to re-use it. There are many options, such as mounting a hard disk, using a USB device, etc. If you have an Internet connection configured, you can email it to yourself, as is illustrated in the next shot:



```
user@pelican:/home$ sudo -s
root@pelican:/home# tar cfz user.tar.gz ./user/
root@pelican:/home#
```

Location: http://mail.google.com/mail/h/gq6a8fdfrzzn/?v=b&pv=tl&cs=b

To: michael.creel@somewhere

Cc:

Bcc:

Subject: archive of PelicanHPC work

Attachments:

/home/user.tar.gz

Dear Michael - here's that file you needed.
Cheers, Michael

If you use PelicanHPC for serious work, it is very convenient to mount a storage device to use as /home, so that your work will be saved between sessions without taking any special steps. When you boot up the frontend node, you have the option to select a storage device to use. This is a feature for advanced users, and I strongly advise that you dedicate a hard disk partition for use with PelicanHPC. If you use a partition with other data on it, you should make sure to back it up before using it with PelicanHPC! Only ext2 and ext3 formats are known to work. This feature has been tested using a very limited set of hardware, so use it with caution. There is also the option to automatically mount a volume that has a special name. See pelican_config in /home/user. This is the best solution for users who want to use PelicanHPC on a long term basis.

Return to contents

Using the make_pelican script

The distributed ISO images provide a bare bones cluster setup system, plus some packages that I use in my research and teaching. There are a few examples taken from my work, which may be of interest to those learning the basics of MPI, or to people interested in econometrics. However, many users will find that Pelican does not contain packages that they need. If one uses pelican_config properly, it is possible to give all nodes of the cluster internet access through the connection of the frontend node, so packages can be simply added using "apt-get". Nevertheless, some users will prefer to have a custom version of the CD image. PelicanHPC is made by running a single script "make_pelican" (with a version number appended), which is available on the download page, and also on the released images. If you have the prerequisites for running the script, it is very easy to make a customized version of Pelican. *The prerequisites are installed on PelicanHPC, so you can build a custom version using the released version.* The prerequisites are:

- an installed version of GNU/Linux. This can be a minimal installation in a chroot jail or a virtual machine running under Linux, Windows or MacOS. If you use a virtual machine, make sure to allocate several GB of disk space.
- the live-build package, version 2.x. Use the version available in Debian Stable: get it at http://packages.debian.org/squeeze/all/live-build/download. It is available as a .deb, and also as source code for use with other distributions. You also need the debootstrap, wget and rsync packages.

To use the script:

1. examine the make_pelican script, which contains some self-explanatory comments. Add the packages you need to the package list section.
2. there is a part following the ##### pelicanhome #### comment where pelicanhome.tar.bz2 is downloaded. This part adds content related to my research and teaching, including some non-Debian software packages. You can easily modify this part to install your own custom content. Get the file pelicanhome.tar.bz2 and examine the contents for examples of how to install non-Debian software.
3. you need to run make_pelican as the root user (e.g. "sudo sh make-pelican"). A fast internet connection is helpful, since a lot of packages need to be downloaded. Also, it helps to build the image on a fast, hopefully multicore computer. Parts of the build process are parallelized and will take advantage of multiple cores. Build time for the default configuration on a decent dual core laptop with lot of RAM is less than half an hour.
4. when you are done, there will be a file "pelicanhpc-custom.iso" in the directory from where you ran the script.
5. There is a manual for Debian Live. Please have a look at it before trying to use make_pelican. Additional information is on the Debian Live homepage. This information is the main documentation, since make_pelican is just a script that provides a specific configuration to the Debian Live system of building a live CD image. Also remember that "man live-build", "man lb_config" and "man lb_build" will give you information.